

Claim 20:

1 20. The apparatus of claim 19 wherein said monitor operates as executable
2 code of the dispatcher algorithm used by an instruction processor when said
3 instruction processor seeks a new task wherein a value related to a current level
4 of busyness of said instruction processor using said monitor executable code is
5 stored in an instruction processor busyness data area.

Claim 21:

1 21. The apparatus of claim 20 wherein said monitor evaluates the
2 values stored in said instruction processor busyness data area and generates a
3 level of busyness value of the multiprocessor system there-from, and based on a
4 comparison between said current busyness value for this one instruction
5 processor, produces a transfer affinity value, and wherein said monitor compares
6 said transfer affinity value against a threshold level value to determine whether
7 said another instruction processor will be permitted to steal a task from said one
8 instruction processor-associated switching queue or not.

Claim 22:

1 22. A dispatcher algorithm for use in a multiple instruction processor computer
2 system having at least three levels of memory, said three levels being at least
3 two cache levels, a first of which is accessible directly by a single one of said
4 instruction processors, a mid-level memory being a multiprocessor-accessible
5 cache accessible by at least two of said instruction processors, and a third
6 memory level being a main memory, accessible by all of said instruction
7 processors, wherein tasks are directed to switching queues for cluster
8 assignment on an affinity basis by an executive and said switching queues are
9 maintained and controlled by said dispatcher algorithm, said dispatcher algorithm
10 comprising:

an executable program for assigning affinity of each new task to a one of said clusters executing said executable program, wherein a cluster is a group of instruction processors,

a set of switching queues of substantially the same number as clusters wherein one switching queue is associated with said one of said clusters and a switching queue is also associated with substantially each other of said clusters, said switching queues having code for their operation and a data area wherein said data area is for maintaining a list of tasks for a cluster to accomplish,

a load balancing level matrix for directing said first cluster to steal a task from a switching queue associated with another cluster in accord with a predetermined mapping within said matrix when said first cluster is looking for an additional task.

Claim 23:

23. A dispatcher algorithm for use in a multiple instruction processor computer system having hierarchical levels of memory, wherein tasks are directed to switching queues for assignment to a processor unit on an affinity basis by an executive and said switching queues are maintained and controlled by said dispatcher algorithm, said dispatcher algorithm comprising:

an executable program for assigning affinity of each new task to a one of said processor units executing said executable program,

a set of switching queues of substantially the same number as processor units wherein one switching queue is associated with one of said processor units and a switching queue is also associated with substantially each other of said processor units, said switching queues having code for their operation and a data area wherein said data area is for maintaining a list of tasks for an processor unit to accomplish,

a load balancing level matrix for enabling said first processor unit to steal a task from a switching queue associated with another processor unit in accord

- 16 with a predetermined mapping within said matrix when said first processor unit is
17 looking for an additional task.

Claim 24:

- 1 24. A multiple instruction processor computer system having a dispatcher
2 algorithm as set forth in claim 23.

Claim 25:

- 1 25. A dispatcher algorithm as set forth in claim 23, wherein each said
2 processor unit comprises either a cluster of instruction processors or a single
3 instruction processor.

Claim 26:

- 1 26. A dispatcher algorithm as set forth in claim 25, wherein at least one of said
2 processor units is comprised of a different number of instruction processors than
3 at least one other one of said processor units.

1